Issues that could be improved

1. table detection model

Layoutparser is not the most accurate model for table detection, if we could find a more powerful model, it would substantially improve the performance of our application. I noticed that [PP-StructureV2] claims in its readme that it is more accurate and faster than layoutparser.

(https://github.com/PaddlePaddlePaddleOCR/blob/release/2.6/ppstructure/docs/PP-StructureV2 introduction.md)

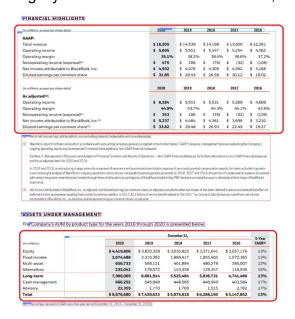
On the PubLayNet dataset, the performance comparison with other methods is shown in the following table. It can be seen that compared to the layoutparser, a layout analysis tool based on Detectron2, our model is about 5% more accurate and predicts about 69 times faster.

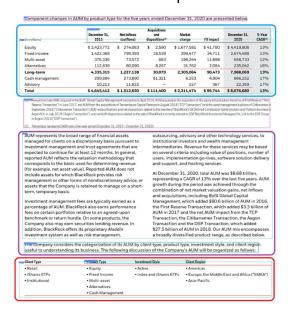


So, I tried to use PP-Structure to detect the tables in pages 8 to 18 of "blackrock-2020-annual-report.pdf". The result shows that it can detect more tables than layoutparser, but it will also mistake some paragraphs for tables.

In this screenshot of page 16, the table boxed by the red rectangle were detected by both methods, blue one was only detected by PP-Structure, and purple is where PP-Structure mistake text for tables.

If we look closely, we can see that there are also some light blue rectangles, they are the areas that PP-structures consider as tables. We can find that most of them are slightly smaller than the real table area, and that will cause extraction problem.





Therefore, the choice depends on our needs. If we want to find as many table as possible, we should choose PP-Structure. Otherwise, the layoutparser is a better choice if we want the extracted tables have a higher accuracy rate.

2. draw lines between rows and columns in borderless tables

PP-structure sometimes has trouble extracting borderless tables, so I was wondering if drawing split lines for the table in advance and then extracting the pre-processed table image could improve the extraction results.

Unfortunately, I didn't find many ways to add borders to the table. cv2 have function like findContours(), However, it is not as accurate as the SLANet model used by PP-structure.

As we can see, this function cannot recognize the original border or cells that occupy many rows, it is just adding split lines between every two rows of objects.

	Snares Issued				Snares Outstanding		
	Treasury						
	Common Shares	Common Shares	Series B Preferred	Series C Preferred	Common Shares	Series B Preferred	Series C Preferred
December 31, 2017	171,252,185	(11,275,070)	823,188	246,522	159,977,115	823,188	246,522
Shares repurchased	_	(3,511,603)	-	_	(3,511,603)	-	
Net issuance of							
common shares							
related to employee							
stock transactions	_	1,087,989	-	_	1,087,989	_	-
PNC LTIP capital							
contribution		19—61		(103,064)	_	·—	(103,064)
December 31, 2018	171,252,185	(13,698,684)	823,188	143,458	157,553,501	823,188	143,458
Shares repurchased		(4,018,905)	_	=	(4,018,905)	_	-
Net issuance of							
common shares							
related to employee							
stock transactions	=	841,184	-	-	841,184	0.	_
PNC LTIP capital							
contribution			[-]	(143,458)			(143,458)
December 31, 2019	171,252,185	(16,876,405)	823,188		154,375,780	823,188	1
Shares repurchased	_	(3,445,554)	-	-	(3,445,554)		_
Net issuance of							
common shares							
related to employee							
stock transactions	_	779,471	-	-	779,471	· -	1.—
Exchange of preferred							
shares series B for							
common shares	823,188	· -	(823,188)	_	823,188	(823,188)	i —
December 31, 2020	172,075,373	(19,542,488)	_	_	152,532,885	_	_

Even worse, sometimes it fails to detect rows and consider the entire table as one single row.

(in millions)	2020	2019	2018
Beginning balance	\$(571)	\$(691)	\$(432)
Foreign currency translation adjustments ⁽¹⁾	234	120	(253)
Reclassification as a result of adoption of accounting guidance			(6)
guidance	-		(6)
Ending balance	\$(337)	\$(571)	\$(691)

In fact, PP-structure's recognition result of cells is a combination of the coordinates, recognition result of a single row, and the coordinates of the cells, and it may be hard to find a better open source "line-drawer", if even PP-structure cannot recognize the cells well.

But if we can find a more accurate table recognizer, it will greatly improve the performance of our program.